This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method of reconciling a first data structure with a second data structure~~ that is a subsequently modified version of the first data structure~~, comprising:

accessing each node in the first data structure for which a change has been made to a corresponding node in the second data structure; and

for each accessed node,

determining if the change made to the corresponding node in the second data structure creates a collision with the first data structure; and

if the change made to the corresponding node in the second data structure does not create a collision with the fast data structure, making the change to the accessed node in the first data structure.

2. (Currently Amended) The method recited in claim 1, further comprising:

determining that the change made to the corresponding node in the second data structure creates a collision with the first data structure; and

determining whether the collision is a mandatory collision or a discretionary collision.

3. (Original) The method recited in claim 2, further comprising:

determining that the change made to the corresponding node in the second data structure creates a discretionary collision with the first data structure, and

making the change to the accessed node in the first data structure.

4. (Original) The method recited in claim 2, further comprising:

determining that the change made to the corresponding node in the second data structure creates a discretionary collision with the first data structure;

determining whether the discretionary collision will allow the change made to the corresponding node in the second data structure to be made to the accessed node; and

if the discretionary collision will allow the change made to the corresponding node in the second data structure to be made to the accessed node, making the change to the accessed node.

5. (Currently Amended) ~~The method recited in claim 1, further comprising:~~

~~employing a log of changes to the second data structure to determine if a change made to a corresponding node in the second data structure creates a collision with the first data structure~~ The method recited in claim 2, further comprising determining whether a collision is discretionary based upon interface rules for the first data structure.

6. (New) The method recited in claim 1, further comprising:

employing a log of changes to the second data structure to determine, for each accessed node in the first data structure, if a change made to a corresponding node in the second data structure.

7. (New) A method of reconciling a first data structure with a second data structure, comprising:

determining which node of the second data structure includes a change from a corresponding node in the first data structure;

for each node in the second data structure including a change,

attempting to access the corresponding node in the first data structure;

if the corresponding node in the first data structure can be accessed, determining, if the change to the second data structure creates a mandatory collision or a discretionary collision,

if the change to the second data structure creates a discretionary collision,

determining if the discretionary collision is forbidden by collision criteria, and

if the discretionary collision is not forbidden by the collision criteria, making the change to the corresponding node in the first data structure.

8. (New) The method recited in claim 7, further comprising deleting empty nodes from the first data structure.

9. (New) The method recited in claim 7, further comprising identifying nodes in the first data structure for with a change to the second data structure creates a collision to a software application maintaining the first data structure.

10. (New) The method recited in claim 7, wherein the collision criteria:

prohibits ink strokes from being added to a leaf node below a pinned node,

prohibits ink strokes from being removed from a leaf node below the pinned node,

prohibits adding leaf nodes below the pinned node,

prohibits removing leaf nodes below the pinned node, and

prohibits re-parenting of leaf nodes below the pinned node.

11. (New) The method recited in claim 7, wherein the collision criteria:

allows late ink strokes to be added to a leaf node below a pinned node under specified conditions,

prohibits ink strokes from being removed from a leaf node below the pinned node,

prohibits adding leaf nodes below the pinned node,

prohibits removing leaf nodes below the pinned node, and

prohibits re-parenting of leaf nodes below the pinned node.

12. (New) The method recited in claim 7, wherein the collision criteria:

allows ink strokes to be added to a leaf node below a pinned node under specified conditions,

prohibits ink strokes from being removed from a leaf node below the pinned node,

prohibits adding leaf nodes below the pinned node,

prohibits removing leaf nodes below the pinned node, and

prohibits re-parenting of leaf nodes below the pinned node.

13. (New) The method recited in claim 7, further comprising determining whether a collision is mandatory based upon interface rules for the first data structure.

14. (New) The method recited in claim 7, further comprising:

employing a log of changes to the second data structure to determine, for each accessed node in the first data structure, if a change has been made to a corresponding node in the second data structure.

15. (New) A method of reconciling a first data structure with a second data structure, comprising:

creating a first copy and a second copy of at least a portion of an original version of the first data structure;

modifying the second copy in a document analysis process to create the second data structure;

receiving at least a portion of a second version of the first data structure;

mapping each node in the second data structure to a corresponding node in the first copy;

analyzing each node in the at least a portion of the second version of the first data structure to determine if the node includes changes from a corresponding node in the first copy; and

if an analyzed node includes changes from a corresponding node in the first copy, adding the corresponding node to a list of nodes for which changes will not be propagated to the first data structure.

16. (New) The method recited in claim 15, further comprising:

analyzing each node in the first copy to determine if a corresponding node is deleted from the second version of the first data structure; and

if the corresponding node is deleted from the second version of the first data structure, adding the corresponding node to a list of nodes for which changes will not be propagated to the first data structure.

17. (New) The method recited in claim 16, further comprising:

analyzing each node in the second data structure, and

if the analyzed node is not in the list of nodes for which changes will not be propagated to the first data structure, propagating changes in the analyzed node from a corresponding node in the second data structure to the corresponding node in the second version of the first data structure.

18. (New) The method recited in claim 16, further comprising:

identifying nodes in the second data structure corresponding to nodes in the second version of the first data structure that have been deleted from the second version of the first data structure;

adding the identifying nodes to a list of deleted nodes; and

deleting identified nodes after all changes have been propagated to the second version of the first data structure.